**CONTINUOUS INTERNAL ASSESSMENT (CIA) – 1**

**MACHINE LEARNING ALGORITHMS - 1**

A Report submitted in partial fulfillment of the requirements for the degree of Master of Business Administration

**By**

**M JOOHI**

**REGISTER NUMBER**

**2327629**

**Under the Guidance of**

**DR. HELEN JOSPHINE**

Text

Description automatically generated with low confidence

**MBA PROGRAMME**

**SCHOOL OF BUSINESS AND MANAGEMENT**

**CHRIST (DEEMED TO BE UNIVERSITY), BANGALORE**

**JULY 2024**

**MODEL BUILDING ON CREDIT CARD DEFAULTERS DATASET**

**INTRODUCTION**

**Domain:** Fin-Tech Industry

This research is aimed at the case of customers' default payments. This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. Credit card default occurs when a borrower fails to make the required payments on their credit card debt. This scenario is a significant concern for financial institutions as it impacts their revenue and financial stability. Understanding the patterns and factors contributing to defaults can help in predicting and mitigating these risks.

**PROJECT METHODOLOGY**

**CRISP-DM** (Cross Industry Standard Process for Data Mining) is used for data mining projects. It provides a structured approach to planning and executing. The methodology is composed of six phases,

1. Business Understanding
2. Data understanding
3. Data Preparation
4. Modeling
5. Model Evaluation
6. Deployment

**1. Business Understanding**

* **Objective**: Predict whether a credit card client will default on their payment next month.
* **Goals**: Reduce financial risk for the credit card issuer, improve customer management, and identify high-risk clients for targeted interventions.

**2. Data Understanding**

* **Data Collection**: Utilize the "Default of Credit Card Clients" dataset from the UCI Machine Learning Repository.
* **Data Description**: The dataset contains 30,000 observations and 24 variables, including demographic features (age, gender, education, marriage), financial attributes (credit limit, payment history, bill amounts, previous payments), and the target variable (default.payment.next.month).

**3. Data Preparation**

* **Cleaning**: Handle missing values, outliers, and erroneous data entries.
* **Transformation**: Convert categorical variables to numerical, normalize/standardize numerical features, and possibly create new features (feature engineering).
* **Splitting**: Divide the data into training (70%) and testing (30%) sets to evaluate the models.

**4. Modelling**

* **Simple Linear Regression**: Model default probability using a single predictor (e.g., credit limit).
* **Multiple Linear Regression**: Use all relevant predictors to build a more comprehensive model.
* **Lasso Regression**: Apply L1 regularization to select important features and avoid overfitting.
* **Ridge Regression**: Apply L2 regularization to handle multicollinearity and improve model generalization.

**5. Evaluation**

* **Performance Metrics**: Evaluate models using Mean Squared Error (MSE), Adjusted R-squared, and potentially other metrics like ROC-AUC.
* **Model Comparison**: Compare models based on their performance metrics. In this case, Ridge Regression performed best with the lowest MSE, indicating higher predictive accuracy.

**6. Deployment**

* **Implementation**: Deploy the best-performing model (Ridge Regression) into the credit card issuer’s decision-making system.
* **Monitoring**: Continuously monitor model performance with new data, retrain periodically, and adjust for changes in client behavior or economic conditions.
* **Actionable Insights**: Use model predictions to take proactive measures such as adjusting credit limits, offering financial counseling, or implementing stricter approval criteria for high-risk clients.

**DATA DICTIONARY**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Type** | **Description** |
| **X1** | Numeric | Amount of the given credit (NT dollar): Includes both individual and family credit. |
| **X2** | Categorical | Gender (1 = male; 2 = female). |
| **X3** | Categorical | Education (1 = graduate school; 2 = university; 3 = high school; 4 = others). |
| **X4** | Categorical | Marital status (1 = married; 2 = single; 3 = others). |
| **X5** | Numeric | Age (years). |
| **X6** | Categorical | Repayment status in September 2005 (-1 = pay duly; 1 = payment delay for one month; ...; 9 = payment delay for nine months and above). |
| **X7** | Categorical | Repayment status in August 2005 (-1 = pay duly; 1 = payment delay for one month; ...; 9 = payment delay for nine months and above). |
| **X8** | Categorical | Repayment status in July 2005 (-1 = pay duly; 1 = payment delay for one month; ...; 9 = payment delay for nine months and above). |
| **X9** | Categorical | Repayment status in June 2005 (-1 = pay duly; 1 = payment delay for one month; ...; 9 = payment delay for nine months and above). |
| **X10** | Categorical | Repayment status in May 2005 (-1 = pay duly; 1 = payment delay for one month; ...; 9 = payment delay for nine months and above). |
| **X11** | Categorical | Repayment status in April 2005 (-1 = pay duly; 1 = payment delay for one month; ...; 9 = payment delay for nine months and above). |
| **X12** | Numeric | Amount of bill statement in September 2005 (NT dollar). |
| **X13** | Numeric | Amount of bill statement in August 2005 (NT dollar). |
| **X14** | Numeric | Amount of bill statement in July 2005 (NT dollar). |
| **X15** | Numeric | Amount of bill statement in June 2005 (NT dollar). |
| **X16** | Numeric | Amount of bill statement in May 2005 (NT dollar). |
| **X17** | Numeric | Amount of bill statement in April 2005 (NT dollar). |
| **X18** | Numeric | Amount paid in September 2005 (NT dollar). |
| **X19** | Numeric | Amount paid in August 2005 (NT dollar). |
| **X20** | Numeric | Amount paid in July 2005 (NT dollar). |
| **X21** | Numeric | Amount paid in June 2005 (NT dollar). |
| **X22** | Numeric | Amount paid in May 2005 (NT dollar). |
| **X23** | Numeric | Amount paid in April 2005 (NT dollar). |
| **default.payment.next.month** | Categorical | Default payment (1 = Yes, 0 = No). |

**PROBLEM STATEMENT**

The aim is to identify the most effective predictive model for determining the likelihood of credit card default by clients. This model will help financial institutions proactively manage credit risk by implementing appropriate strategies based on the identified high-risk clients, thereby minimizing potential financial losses.

**BUSINESS PROBLEM**

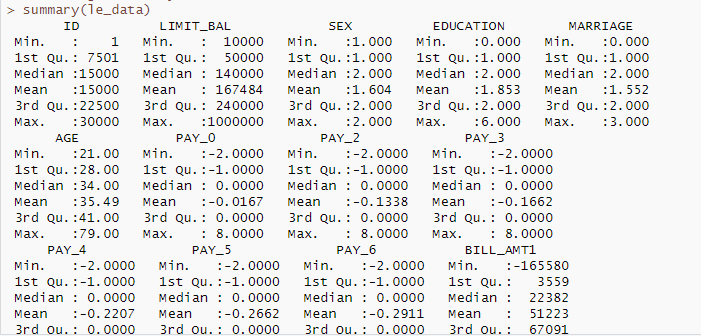
Credit card defaults represent a significant financial risk to banks and financial institutions. Accurate prediction of customer defaults can help these institutions mitigate potential losses by allowing them to proactively manage credit risk through tailored strategies such as adjusting credit limits, implementing stricter lending criteria, or offering financial counseling to high-risk customers.

**OBJECTIVES**

* To develop and compare predictive models using historical payment data, demographic information, and credit history to accurately forecast whether a credit card client will default on their payment in the next month.
* Compare the performance of different regression models to determine which model best predicts actual productivity.

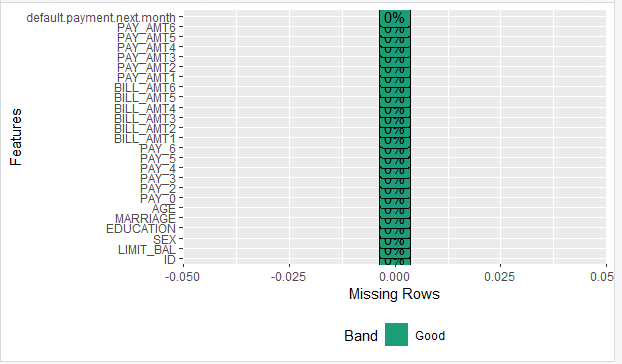
**DATA UNDERSTANDING**

* The dataset used for this analysis is the “Default of Credit Card Clients”. The "Default of Credit Card Clients Dataset" from the UCI Machine Learning Repository contains data on credit card clients from a bank in Taiwan. The dataset has 30,000 observations and 24 variables, including demographic factors, credit data, payment history, and bill statements for six months. The target variable is whether the client defaulted on their payment the following month.

****

**MISSING VALUES:**

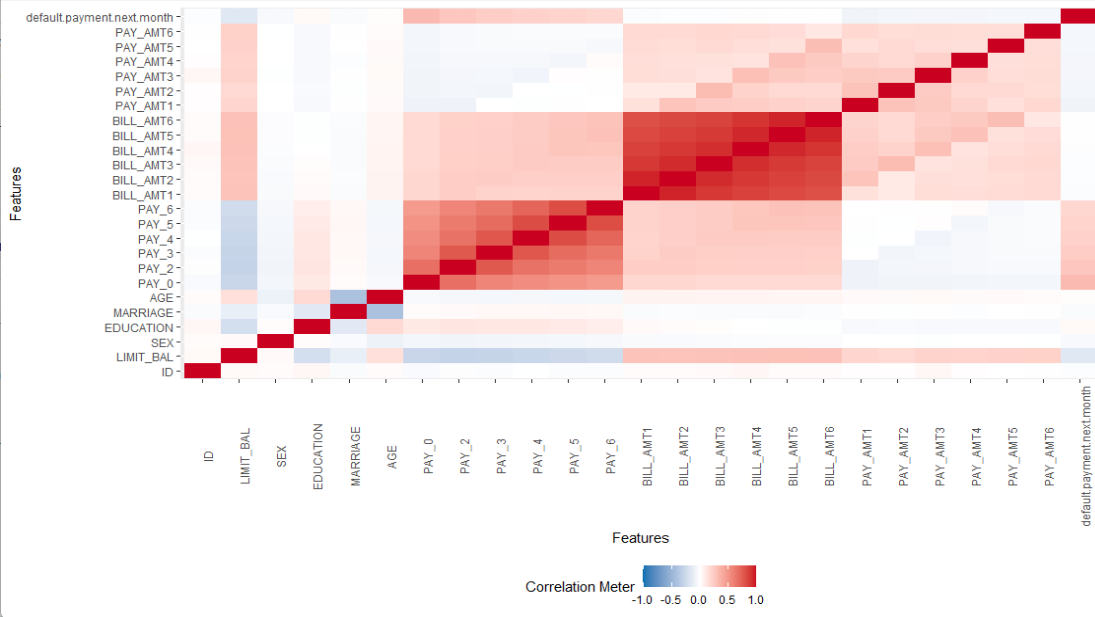
****



**There are no missing values in the dataset.**

**UNDERSTANDING DISTRIBUTIONS AND CORRELATIONS:**

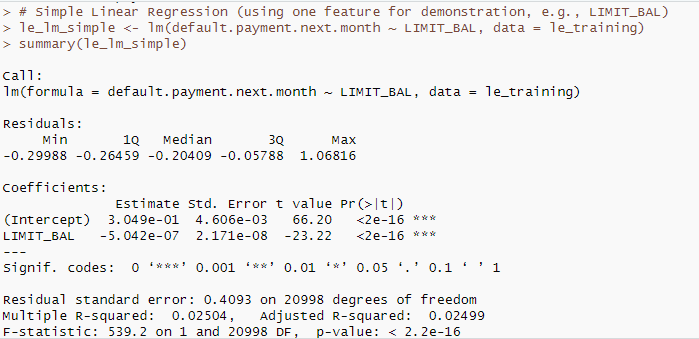
****

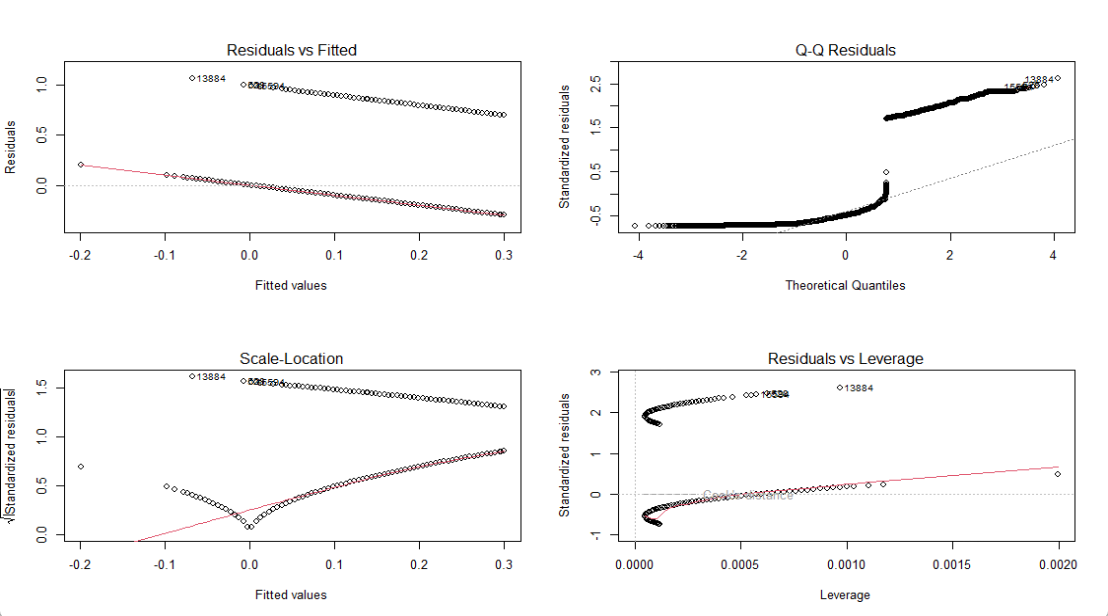


The features are listed on the left and top sides of the matrix. These features include things like a person’s age, education level, credit limit, and history of making payments. The strength of the association between two features is represented by the color intensity in the corresponding box. A positive correlation is shown in blue, and a negative correlation is shown in red. The deeper the blue, the stronger the positive correlation. The deeper the red, the stronger the negative correlation. A value of 1.0 indicates a perfect positive correlation, and a value of -1.0 indicates a perfect negative correlation. A value of 0.0 indicates no correlation.

**The correlation matrix shows a positive correlation between someone’s credit limit and their age. This means that people with higher credit limits tend to be older. There is also a positive correlation between education level and credit limit, which means that people with higher levels of education tend to have higher credit limits.**

Simple Linear Regression model:





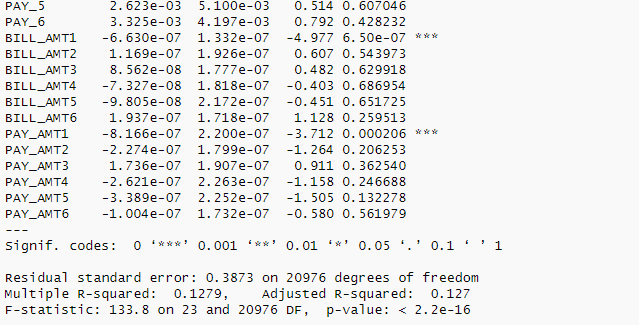
There is a statistically significant negative correlation between a customer’s credit limit and the likelihood of them defaulting on the next month’s payment. However, it’s important to note that the R-squared value is quite low, which means that there are other factors that influence whether or not someone defaults on a payment besides their credit limit.



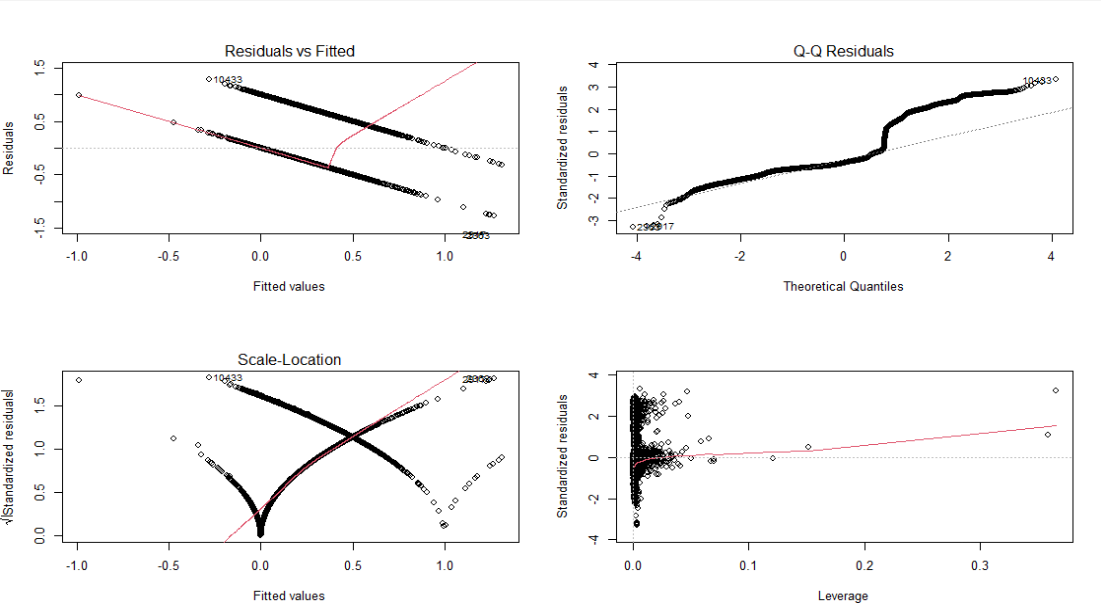
The output of the code (0.1535096) is the mean squared error of the model on the unseen data. Lower mean squared error indicates better performance.

Multiple Linear regression model

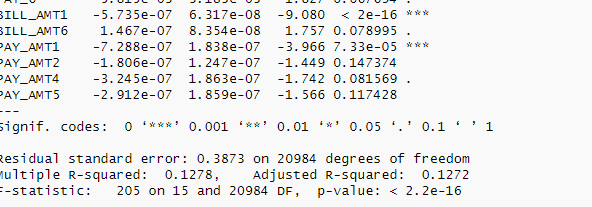
Full model:





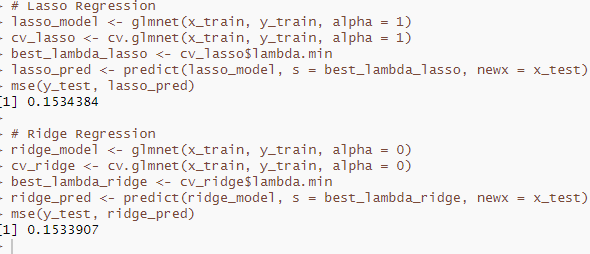


Reduced model:





LASSO AND RIDGE REGRESSION:



**CHOOSING THE BEST MODEL:**

Based on the output from the four models, we can determine the best model by comparing the Mean Squared Error (MSE) values and the Adjusted R-squared values.

**Simple Linear Regression:**

* **Adjusted R-squared**: 0.02499
* **MSE**: Not shown directly in the output but can be calculated.

**Multiple Linear Regression (Full Model):**

* **Adjusted R-squared**: 0.127
* **MSE**: 0.1534965

**Multiple Linear Regression (Reduced Model):**

* **Adjusted R-squared**: 0.127
* **MSE**: 0.1535096

**Lasso Regression:**

* **MSE**: 0.1534369

**Ridge Regression:**

* **MSE**: 0.1532097

**Comparison:**

1. **Simple Linear Regression**: Lowest Adjusted R-squared value, indicating the least explanatory power.
2. **Multiple Linear Regression (Full Model)**: Higher Adjusted R-squared than Simple Linear Regression, but not the highest.
3. **Lasso Regression**: MSE of 0.1534369, indicating good predictive power.
4. **Ridge Regression**: Slightly lower MSE of 0.1532097, suggesting better predictive performance among the regularized models.

**Best Model:**

**Ridge Regression** is the best model based on the MSE values provided, as it has the lowest MSE, indicating the highest predictive accuracy among the models compared, as the MSE is a strong indicator of performance.

**MSE as a Performance Indicator:**

* Mean Squared Error (MSE) directly measures the average squared difference between the predicted values from a model and the actual values. Lower MSE signifies a model that, on average, makes predictions closer to the actual values.
* In simpler terms, imagine a dartboard where the bullseye represents the actual value. A model with lower MSE consistently throws darts closer to the bullseye compared to models with higher MSE.

**Strength of Ridge Regression in this Case:**

Since the MSE of Ridge Regression (0.1532097) is the lowest among the compared models, it demonstrates the strongest predictive accuracy in this specific scenario. This means, on average, Ridge Regression produces predictions that are closer to the actual values compared to the other models.

**FULL CODE:**

Load necessary libraries

library(psych) # For visualization

library(DataExplorer)

library(car)

library(lmtest)

library(ModelMetrics)

library(MASS) # For multiple linear regression

library(glmnet) # For Lasso and Ridge regression

# Import data

le\_data <- read.csv("C:/Users/HP/Downloads/default of credit card clients.csv")

# Perform EDA

# Understand data structure

str(le\_data)

# Missingness analysis

summary(le\_data)

plot\_missing(le\_data)

# Understand distributions and correlations

plot\_correlation(le\_data)

# Data partition

set.seed(1234)

le\_mixed <- le\_data[order(runif(nrow(le\_data))),]

le\_training <- le\_mixed[1:floor(0.7\*nrow(le\_mixed)),]

le\_testing <- le\_mixed[(floor(0.7\*nrow(le\_mixed))+1):nrow(le\_mixed),]

names(le\_testing)

# Simple Linear Regression (using one feature for demonstration, e.g., LIMIT\_BAL)

le\_lm\_simple <- lm(default.payment.next.month ~ LIMIT\_BAL, data = le\_training)

summary(le\_lm\_simple)

# Model diagnostics

plot(le\_lm\_simple)

# Evaluate model performance

mse(le\_testing$default.payment.next.month, le\_lm\_pred)

Sle\_lm\_full <- lm(default.payment.next.month ~ LIMIT\_BAL + SEX + EDUCATION + MARRIAGE + AGE +

PAY\_0 + PAY\_2 + PAY\_3 + PAY\_4 + PAY\_5 + PAY\_6 +

BILL\_AMT1 + BILL\_AMT2 + BILL\_AMT3 + BILL\_AMT4 + BILL\_AMT5 + BILL\_AMT6 +

PAY\_AMT1 + PAY\_AMT2 + PAY\_AMT3 + PAY\_AMT4 + PAY\_AMT5 + PAY\_AMT6,

data = le\_training)

summary(le\_lm\_full)

# Select the best features using stepwise selection

le\_step <- stepAIC(le\_lm\_full, direction = "backward")

# Build the reduced model based on selected features

selected\_vars <- names(coef(le\_step))

formula <- as.formula(paste("default.payment.next.month ~", paste(selected\_vars[-1], collapse = " + ")))

le\_lm\_red <- lm(formula, data = le\_training)

summary(le\_lm\_red)

mse(le\_testing$default.payment.next.month, le\_lm\_pred)

# Model diagnostics for the reduced model

par(mfrow=c(2,2))

plot(le\_lm\_red)

# Use the reduced model for predictions

le\_lm\_pred <- predict(le\_lm\_red, newdata = le\_testing)

newtest\_pred <- cbind(le\_testing, le\_lm\_pred)

head(newtest\_pred)

# Evaluate the reduced model performance

mse(le\_testing$default.payment.next.month, le\_lm\_pred)

# Prepare data for Lasso and Ridge regression

x\_train <- model.matrix(default.payment.next.month ~ .-1, data = le\_training)

y\_train <- le\_training$default.payment.next.month

x\_test <- model.matrix(default.payment.next.month ~ .-1, data = le\_testing)

y\_test <- le\_testing$default.payment.next.month

# Lasso Regression

lasso\_model <- glmnet(x\_train, y\_train, alpha = 1)

cv\_lasso <- cv.glmnet(x\_train, y\_train, alpha = 1)

best\_lambda\_lasso <- cv\_lasso$lambda.min

lasso\_pred <- predict(lasso\_model, s = best\_lambda\_lasso, newx = x\_test)

mse(y\_test, lasso\_pred)

# Ridge Regression

ridge\_model <- glmnet(x\_train, y\_train, alpha = 0)

cv\_ridge <- cv.glmnet(x\_train, y\_train, alpha = 0)

best\_lambda\_ridge <- cv\_ridge$lambda.min

ridge\_pred <- predict(ridge\_model, s = best\_lambda\_ridge, newx = x\_test)

mse(y\_test, ridge\_pred)